

Ref. 3

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-302171

(43) 公開日 平成7年(1995)11月14日

(51) Int. Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/06	3 0 2 A			
G 1 1 B 20/10		D 7736-5D		

審査請求 未請求 請求項の数20 O L (全 13 頁)

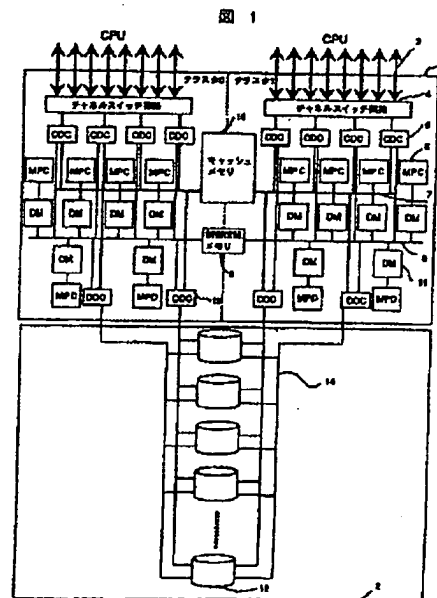
(21) 出願番号	特願平6-98055	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目6番地
(22) 出願日	平成6年(1994)5月10日	(72) 発明者	小町谷 常孝 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
		(72) 発明者	角田 仁 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
		(72) 発明者	加茂 善久 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
		(74) 代理人	弁理士 小川 勝男

(54) 【発明の名称】 ディスク制御装置

(57) 【要約】

【構成】 ディスク制御装置において、従来集中管理されていた制御情報を、マイクロプロセッサ単位に持たせた分散メモリに格納する。

【効果】 制御情報を分散化することで、制御情報へのアクセスネックを解消し、性能向上を可能とした。



1

【特許請求の範囲】

【請求項1】上位装置からのデータの入出力要求に対する、当該データを格納してある、又は格納するディスク装置と、該ディスク装置と上位装置との間のデータの入出力を管理する制御装置において、制御装置内のデータの入出力を管理するマイクロプロセッサが参照および登録を行う制御情報を格納する半導体メモリを、該制御装置内において2個以上に分割することを特徴とするディスク制御装置。

【請求項2】請求項1において、制御装置内のデータの入出力を管理する制御情報を、該制御装置内において2個以上に分割したディスク制御装置。

【請求項3】請求項1において、制御装置内のデータの入出力を管理するマイクロプロセッサが参照および登録を行う制御情報を格納する、該制御装置内において2個以上に分割された半導体メモリの中の、少なくとも1個以上を不揮発化したディスク制御装置。

【請求項4】請求項1または2において、制御装置内のデータの入出力を管理する各マイクロプロセッサに、制御装置内のデータの入出力を管理する制御情報を格納する半導体メモリを設けたディスク制御装置。

【請求項5】請求項1において、制御装置内のデータの入出力を管理する各マイクロプロセッサに設けられた半導体メモリと、制御装置内のデータの入出力を管理する制御情報を格納する半導体メモリの両方を持つディスク制御装置。

【請求項6】請求項5において、制御装置内のデータの入出力を管理する制御情報を格納する半導体メモリ内の一部の制御情報を、制御装置内のデータの入出力を管理する各マイクロプロセッサに設けられた半導体メモリに格納するディスク制御装置。

【請求項7】請求項5において、制御装置内のデータの入出力を管理する制御情報を格納する半導体メモリ内の一部の制御情報のコピーを、制御装置内のデータの入出力を管理する各マイクロプロセッサに設けられた半導体メモリに、格納したディスク制御装置。

【請求項8】請求項4において、制御装置内の、データの入出力を管理する各マイクロプロセッサに設けられた半導体メモリに、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に参照を行う制御情報を格納するディスク制御装置。

【請求項9】請求項8において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、参照を行うマイクロプロセッサに設けられた半導体メモリ内に当該制御情報が存在した場合は、参照を行うマイクロプロセッサに設けられた半導体メモリから、当該制御情報の参照を行うディスク制御装置。

【請求項10】請求項9において、制御装置内のデータ

(2)

特開平7-302171

2

の入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、参照を行うマイクロプロセッサに設けられた半導体メモリ内に当該制御情報が存在しないため、他の半導体メモリから、当該制御情報の参照を行う際に、参照を行うマイクロプロセッサに設けられた半導体メモリ内に、他の半導体メモリから参照した制御情報を格納するディスク制御装置。

【請求項11】請求項4または5において、制御装置内のマイクロプロセッサに設けられた半導体メモリの一部をバッテリーにより不揮発化するディスク制御装置。

【請求項12】請求項1において、制御装置内の半導体メモリに制御情報の格納に関する情報を格納する領域を確保するディスク制御装置。

【請求項13】請求項11において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、マイクロプロセッサに設けられた半導体メモリ内に制御情報の登録を行うディスク制御装置。

【請求項14】請求項13において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、マイクロプロセッサに設けられた不揮発化された半導体メモリ内に制御情報の登録を行うディスク制御装置。

【請求項15】請求項12において、制御装置内の半導体メモリに格納されている、制御情報の格納に関する情報により、制御装置内のデータの入出力を管理するマイクロプロセッサが制御情報の格納先を判定するディスク制御装置。

【請求項16】請求項13において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、登録を行うマイクロプロセッサに設けられた半導体メモリ内に当該制御情報が存在した場合は、登録を行うマイクロプロセッサに設けられた半導体メモリに、当該制御情報の登録を行うディスク制御装置。

【請求項17】請求項13において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、登録を行うマイクロプロセッサに設けられた半導体メモリ内に当該制御情報が存在しないため、他の半導体メモリに、当該制御情報の登録を行う際に、登録を行うマイクロプロセッサに設けられた半導体メモリ内に、当該制御情報を登録するディスク制御装置。

【請求項18】請求項17において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、登録を行うマイクロプロセッサに設けられた半導体メモリと、他の半導体メモリの両方に、当該制御情報の登録を行うディスク制御装置。

【請求項19】請求項14において、制御装置内のデータ

3

タの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、該マイクロプロセッサに設けられた不揮発化された半導体メモリ内から、該マイクロプロセッサに設けられた不揮発化されていない半導体メモリ内へ、制御装置内のデータの入出力を管理するために必要な制御情報の移動を行うディスク制御装置。

【請求項20】請求項19において、制御装置内のデータの入出力を管理するマイクロプロセッサが、制御装置内のデータの入出力を管理する際に、該マイクロプロセッサに設けられた不揮発化されていない半導体メモリ内から、該マイクロプロセッサに設けられた不揮発化された半導体メモリ内へ、制御装置内のデータの入出力を管理するために必要な制御情報の移動を行うディスク制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はコンピュータシステムに係り、特に高性能な入出力動作を可能とするディスクシステムに関する。

【0002】

【従来の技術】通常のコンピュータシステムにおいては、CPU等の上位側が必要とするデータは2次記憶装置に格納され、CPUが必要に応じて2次記憶装置に対してデータの書き込み、読みだしを行っている。この2次記憶装置としては一般に不揮発な記憶媒体が使用され、代表的なものとして磁気ディスク装置(以下ドライブとする)、光ディスクなどがあげられる。このような2次記憶装置では、一般にCPUとドライブ間のデータ転送制御を行うディスク制御装置(以下DKCとする)が存在する。

【0003】ディスク制御装置の一例を以下に説明する。DKC1では図7に示すようにチャネルデータ制御回路(CDC)5とドライブデータ制御回路(DDC)13によりデータ転送は実行される。チャネルデータ制御回路(CDC)5はCPUとDKC1内のキャッシュメモリ10間のデータ転送を行い、ドライブデータ制御回路(DDC)13はキャッシュメモリ10とドライブ12間のデータ転送を行う。CDC5、キャッシュメモリ10、DDC13はデータ用の共通バス(DBS)7に接続される。DKC1におけるデータ転送制御は、各CDC5、DDC13の制御を行うマイクロプロセッサ(MP)6の管理のもとで行われる。各MP6は制御信号用の共通バス(CBS)8に接続されている。

【0004】この様にDKC1内では各MP6がマルチプロセッサ構成になっており、それぞれのMP6がバス、キャッシュメモリ10、ドライブ12等の共有の資源を効率良く使用することで、スムーズなデータ転送を実現している。このように、MP6が協調して動作するために、DKC1内では、CBS8を介して制御情報メ

(3)

特開平7-302171

4

モリ(CM)9内の制御情報の参照および更新を繰り返す必要がある。従来のDKC1ではこのように制御情報はCM9にまとめられ、一括管理されている。

【0005】

【発明が解決しようとする課題】従来のDKC1では各MP6が協調して動作するため、1回の入出力処理においてCBS8を介して何回もCM9にアクセスを行う必要がある。このため、今後DKC1内のMP6の数が増加したり、MP6の処理性能が向上すると、CBS8、CM9間がネックになる。

【0006】

【課題を解決するための手段】上記課題を解決するため本発明は、ディスク制御装置内の各MP6に分散メモリ(DM)11を設け、CM9内の制御情報を分散させる。

【0007】

【作用】この様に各MP6にDM11を設け、CM9内のデータを分散させると、MP6が必要とする制御情報がDM11内に存在すれば、CM9へのアクセスが必要でなくなるため、CBS8、CM9間のネックが解消される。

【0008】

【実施例】

(実施例1)以下本発明の一実施例を図1により説明する。図1は本発明のDKC1内の構造を示す。図1に示すようにDKC1はキャッシュメモリ10、制御情報メモリ9をはさんでクラスタ0、1は対称になっている。DKC1は実際にCPUとドライブ12間でデータ転送を行うデータ転送系と、このデータ転送を管理する制御系で構成される。クラスタ0、1は全く同じ構成になっているため、以下の説明はクラスタ0について行う。

【0009】まず、DKC1内のデータ転送系について説明する。まず、CPUとDKC1間のデータ転送方法を示す。CPUとDKC1はチャネルバス3により接続され、このチャネルバス3はDKC1内においてチャネルスイッチ回路4に接続されている。チャネルスイッチ回路4はDKC1内において同時に動作可能なチャネルバス3は最大4本のため、4本以上のチャネルバス3の中から、最大4本を選択する回路である。各チャネルスイッチ4には各々4個ずつチャネルデータ制御回路(CDC)5が接続されている。このCDC5はデータ用共通バス(DBS)7に接続されている。DBS7にはキャッシュメモリ10も接続されている。

【0010】CPUから転送されてきたデータはCDC5の制御のもと、DBS7を通過してキャッシュメモリ10に格納される。また、キャッシュメモリ10からCPUにデータを転送する場合は、逆にキャッシュメモリ10からデータを読みだし、DBS7を通過してCDC5の制御のもと、CPUへ転送される。このように、CDC5はCPUとDKC1間のデータ転送を制御する回路で

5

ある。

【0011】次に、DKC1とドライブ12間のデータ転送方法を示す。DBS7にはCDC5、キャッシュメモリ10以外にも2個のドライブデータ制御回路(DDC)13が接続されている。ドライブ12から転送されてきたデータはDDC13の制御のもと、DBS7を通過してキャッシュメモリ10に格納される。また、キャッシュメモリ10からドライブ12にデータを格納する場合は、逆にキャッシュメモリ10からデータを読みだし、DBS7を通過してDDC13の制御のもと、ドライブ12へ転送される。このように、DDC13はCPUとドライブ12間のデータ転送を制御する回路である。以上のようにDKC1ではCDC5、DDC13によりCPU、キャッシュメモリ10、ドライブ12間のデータ転送を行う。

【0012】次に、上記のデータ転送を制御する制御系について説明する。DKC1の各クラスには6個のマイクロプロセッサ(MP)6が各々分散メモリ(DM)11を介して制御信号用の共通バス(CBS)8に接続されている。また、このCBS8には制御情報メモリ(CM)9が接続されている。MP6にはCDC5を制御するMPCとDDC13を制御するMPDがある。

【0013】MPCはそれぞれ制御を担当するCDC5が決まっており、MPCは制御情報を参照し、この参照結果をもとにCDC5に対し、CPU、キャッシュメモリ10間のデータ転送を指示する。

【0014】具体的には図2に示すように、CPUからDKC1に対しデータの書き込み要求が発行された場合、この要求を受け付けたMPCはキャッシュメモリ10へのデータ転送が可能かどうかを調べるために制御情報を参照に行く(15)。この時、当該MP1はDM11内に当該制御情報が格納されているかどうかを調べる(16)。当該MPCのDM11内に当該制御情報が格納されている場合(ヒット)は、当該DM11内の当該制御情報に対し参照を行う(21)。

【0015】制御情報の参照により、キャッシュメモリ10が使用可能な場合、CBS8が使用可能な場合は(18)、当該MP1はCM9内の制御情報に対しキャッシュメモリ10の使用を登録する(19)。もしこの時CBS8の使用が不可の場合はCBS8が使用可能になるまで待ち(20)、バスが空き次第、CM9内の制御情報に対しキャッシュメモリ10の使用を登録する(19)。この時、他のDM11に同一の制御情報が格納されている場合は、参照、登録を行う当該MP1は同一制御情報が格納されている他のDM11内の当該制御情報を無効にする(26)。

【0016】一方、当該MPCのDM11内に当該制御情報が格納されていない場合(ミス)は、当該MPCはCBS8が使用可能かどうかを調べ(22)、使用可能な場合はCM9内の当該制御情報を参照し、キャッシュ

(4)

特開平7-302171

6

メモリ10が使用可能な場合は、当該MPCはCM9内の制御情報に対しキャッシュメモリ10の使用を登録する(23)。もし、CBS8の使用が不可の場合は、CBS8が使用可能になるまで待ち(24)、バスが空き次第、CM9内の制御情報を参照し、キャッシュメモリ10が使用可能な場合は、当該MPCはCM9内の制御情報に対しキャッシュメモリ10の使用を登録する(23)。この時、他のDM11に同一の制御情報が格納されている場合は、参照、登録を行う当該MPCは同一制御情報が格納されている他のDM11内の当該制御情報を無効にする。

【0017】この様に、キャッシュメモリ10の使用が登録された後に、MPCはCPUに対しデータ転送の許可を報告する。MPCからのデータ転送許可を受け取ったCPUはDKC1にデータを転送する。DKC1がCPUからデータを受け取った後は、先のデータ転送系で説明した手順でキャッシュメモリ10にデータを格納する。

【0018】次にこの様にキャッシュメモリ10に格納されたデータをドライブ12に格納する場合を説明する。キャッシュメモリ10に格納されたデータをドライブ12に格納する場合は、MPCからMPDに制御権が移行する。MPDもそれぞれ制御を担当するDDC13が決まっており、MPDは制御情報を参照し、この参照結果をもとにDDC13に対し、キャッシュメモリ10、ドライブ12間のデータ転送を指示する。

【0019】具体的には、MPCからキャッシュメモリ10からドライブ12に対しデータの書き込みをMPCから依頼されたMPDは、キャッシュメモリ10からドライブ12へのデータ転送可能かどうかを調べるために制御情報を参照に行く。この時参照する制御情報はキャッシュメモリ10とDKC1とドライブ12間のドライブバス14とドライブ12の使用状況に関する制御情報である。まずMPDは当該DM11内に当該制御情報が格納されているかどうかを調べる(16)。当該MPCのDM11内に当該制御情報が格納されている場合(ヒット)は、当該DM11内の当該制御情報に対し参照を行う(21)。

【0020】これらの制御情報の参照により、キャッシュメモリ10とドライブバス14とドライブ12が使用可能でデータ転送可能な場合は、CBS8が使用可能な場合は(18)、当該MPDはCM9内の制御情報に対しキャッシュメモリ10とドライブバス14とドライブ12の使用を登録する(19)。もしこの時CBS8の使用が不可の場合はCBS8が使用可能になるまで待ち(20)、バスが空き次第、CM9内の制御情報に対しキャッシュメモリ10とドライブバス14とドライブ12の使用を登録する(19)。この時、他のDM11に同一の制御情報が格納されている場合は、参照、登録を行う当該MP1は同一制御情報が格納されている他のDM1

(5)

特開平7-302171

8

1内の当該制御情報を無効化する(26)。

【0021】一方、当該MPDのDM11内に当該制御情報が格納されていない場合(ミス)は、当該MPDはCBS8が使用可能かどうかを調べ(22)、使用可能な場合はCM9内の当該制御情報を参照し、キャッシュメモリ10とドライブバス14とドライブ12が使用可能な場合は、当該MPDはCM9内の制御情報に対しキャッシュメモリ10とドライブバス14とドライブ12の使用を登録する(23)。もし、CBS8の使用が不可の場合は、CBS8が使用可能になるまで待ち(24)、バスが空き次第、CM9内の制御情報を参照し、キャッシュメモリ10とドライブバス14とドライブ12が使用可能な場合は、当該MPDはCM9内の制御情報に対しキャッシュメモリ10とドライブバス14とドライブ12の使用を登録する(23)。この時、他のDM11に同一の制御情報が格納されている場合は、参照、登録を行う当該MPDは同一制御情報が格納されている他のDM11内の当該制御情報を無効化する。

【0022】この様に、キャッシュメモリ10とドライブバス14とドライブ12の使用が登録された後に、MPDはDDC13に対しキャッシュメモリ10からドライブ12へのデータ転送を指示する。MPDからのデータ転送指示を受け取ったDDC13はキャッシュメモリ10からドライブにデータを転送する。DDC13によるドライブ12へのデータ転送は、先のデータ転送系で説明した手順で行う。

【0023】CPUからDKC1へのデータ書き込みの場合とDKC1からドライブ12へのデータ書き込みの場合のデータ転送を制御する制御系の動作について説明してきた。これとは逆に、ドライブ12からDKC1にデータを読みだし、DKC1からCPUにデータを読み出す場合のデータ転送を制御する制御系の動作は、CPUからDKC1にデータを書き込む場合とDKC1からドライブ12にデータを書き込む場合のデータ転送を制御する制御系の動作と同じになる。

【0024】本実施例では、以上のようにDM11内の制御情報は参照専用である。DM11内には制御情報の中で、比較的参照頻度の高い制御情報を予め格納しておく。制御情報に登録を行う場合は、CBS8を介してCM9に対し行われる。これは、制御情報はDKC1において非常に重要な情報であるため、CM9において確実に管理を行うためである。このため、CM9は停電等の事故により、制御情報が消失しないようにバッテリでバックアップされ、しかも、二重化されている方が望ましい。このように、制御情報がバッテリでバックアップされ、しかも、二重化されているCM9において確実に集中管理されておれば、DM11には揮発半導体メモリを使用し、しかも一重でも構わない。

【0025】(実施例2) 本実施例ではDM11内の制御情報の動的な変更を可能とした例を示す。本実施例に

おけるDKC1内の構成は、実施例1と同様に図1のようになっており、データ転送系の動作も実施例1と同様である。従って以下では制御系の動作を中心に図3を用いて説明する。

【0026】MP6において制御情報を参照または登録する必要が生じた場合(31)、まず、MP6は自分のDM11内に当該制御情報が存在するかどうかを調べる(32)。もし、自分のDM11内に当該制御情報が存在する場合(ヒット)は、その制御情報に対し登録を行うかどうかを認識する(33)。MP6は登録の場合はCBS8の使用が可能かどうかを調べ(34)、使用可能な場合はCM9に登録を行う(35)。この時、他のDM11に同一の制御情報が格納されている場合は、登録を行う当該MP1は同一制御情報が格納されている他のDM11内の当該制御情報を無効化する(36)。一方、CBS8の使用が不可の場合は、CBS8が使用可能になるまで待ち(37)、バスが空き次第、CM9内の制御情報に対し登録し(35)、他のDM11に同一の制御情報が格納されている場合は、登録を行う当該MP1は同一制御情報が格納されている他のDM11内の当該制御情報を無効化する(36)。

【0027】自分のDM11内にヒットし(32)、参照の場合は、実施例1と同様にDM11から、当該制御情報を参照する(38)。

【0028】一方、自分のDM11内に当該制御情報が存在しない場合(ミス)は、ヒット時と同様にまず、その制御情報に対し登録を行うかどうかを認識する(39)。もし登録であればヒット時と同様にCM9に登録を行う(40)、(41)、(42)、(43)。参照の場合はCBS8の使用が可能かどうかを調べ(44)、使用が不可の場合は、CBS8が使用可能になるまで待つ48。使用可能な場合はCM9または他のMP6のDM11から参照した制御情報を、参照しようとしているMP6のDM11内に保持するかしないかを判断する(45)。もし、保持する場合は、CM9または他のMP6のDM11から参照した制御情報はDM11内に格納され(46)、同時に参照される(47)。また、保持しない場合はCM9または他のMP6のDM11から当該制御情報を参照(49)するのみである。

【0029】本実施例では、参照時にミスした場合、CM9より当該制御情報を参照すると同時にDM11内に保持することも可能としたため、DM11内に格納されている制御情報の動的な移動が可能となった。

【0030】本実施例でも実施例1と同様に、DM11内の制御情報は参照専用である。制御情報に登録を行う場合は、CBS8を介してCM9に対し行われる。CM9は停電等の事故により、制御情報が消失しないようにバッテリでバックアップされ、しかも、二重化されている方が望ましい。このように、制御情報がバッテリでバックアップされ、しかも、二重化されているCM9にお

9

いて確実に集中管理されておれば、DM11は揮発半導体メモリを使用し、しかも一重でも構わない。

【0031】(実施例3)本実施例におけるDKC1内の構成は図4のようになっている。本実施例では各DM11に対し、バッテリーによりバックアップされ不揮発化された分散メモリ(NDM)30も持ち、このNDM30に対し登録を行えることを特徴とし、それ以外の構成要素は実施例1と同じである。

【0032】本実施例の制御系の動作を中心に図3および5を用いて以下に説明する。実施例2ではDM11にミスした場合、参照時にはCM9から制御情報をDM11に保持することが可能となった例を示した。本実施例ではこれを更に拡大し、NDM30に対し登録を可能とした。本実施例は図3において登録の処理が異なる。そこで、処理が異なる登録の部分だけを図5に示し以下に説明する。ここで、図5の処理51は図3の処理33に、図5の処理54は図3の処理36に対応している。

【0033】MP6において制御情報を参照または登録する必要が生じた場合(31)、まず、MP6は自分のDM11内に当該制御情報が存在するかどうかを調べる(32)。もし、自分のDM11内に当該制御情報が存在する場合(ヒット)は、その制御情報に対し登録を行うかどうかを認識する(33)。登録の場合、MP6はNDM30内の当該制御情報に対し登録を行うかを判断する(52)。

【0034】この時のMP6の判断方法について以下に図6を用いて説明する。DKC1内の制御情報は図6のアドレスマップに示すように、メモリ空間上のアドレスで管理され制御情報の特性により分散化の可否および、どのように分散化するかが決定される。aaaaからbbbbまでのアドレスで指定される制御情報は分散化しても良いが、不揮発化する必要の有る制御情報(不揮発分散化制御情報)とする。bbbbからccccまでのアドレスで指定される制御情報は分散化しても良いが、不揮発化する必要の無い制御情報(揮発分散化制御情報)とする。ccccからddddまでのアドレスで指定される制御情報は分散化してはいけない制御情報(非分散化制御情報)とする。不揮発分散化制御情報は比較的連続して登録が行われる制御情報とする。登録された制御情報は停電等により消失しては困るため、不揮発化されているメモリに格納する必要がある。また、連続してアクセスされる場合、DM11に格納されていれば、CM9へ参照、登録せずにすむ。また、揮発分散化制御情報は比較的連続して参照が行われる制御情報とする。本実施例では、揮発分散化制御情報は、そのコピーを必ずCM9内に持つておくことを前提とした。

【0035】参照が主な制御情報は、登録による変更が少ないため、CM9内には常にコピーが存在し、停電等により揮発なDM11に格納されている制御情報が消失しても、CM9により回復できる。非分散化制御情報は

(6)

特開平7-302171

10

多数のMP6から、頻繁に参照、更新が行われる制御情報とする。この様に多数のMP6から、頻繁に参照、更新が行われると、更新のたびに他DM11またはCM9内の同一制御情報の無効化が必要になる。この様に無効化された後は、他のMP6が参照、登録した場合は必ずミスとなり、CM9から当該制御情報を参照しなければならぬ。このように、他DM11またはCM9に対し、頻繁に無効化が行われるようなデータはCM9に格納しておき、集中制御する方が分散化するより制御が簡単で、しかも、性能が良い。この様に制御情報の特性に応じて分散化の有無と、方法を決定する。

【0036】MP6はこのアドレスマップにしたがい、登録しようとしている制御情報が不揮発分散化制御情報であれば、NDM30内の当該制御情報に登録すると判断し、登録する(53)。また、揮発分散化制御情報であれば、DM11内の当該制御情報に対し登録すると判断し、登録する(53)。また、非分散化制御情報であれば実施例1、2と同様にCBS8の使用が可能かどうかを調べ(55)、CM9に対し登録を行う(56)。

【0037】一方、ミス時においても登録の場合はヒット時と同様に図5のように登録を行う。

【0038】本実施例でも、実施例2と同様に参照時にミスした場合、CM9より当該制御情報を参照すると同時にDM11またはNDM30内に保持することも可能である。この時は登録と同様に図6のアドレスマップにしたがい、不揮発分散化制御情報か揮発分散化制御情報か非分散化制御情報かを判断する。

【0039】本実施例では実施例1、2と異なり、DM11またはNDM30内の制御情報は参照と登録の両方に使用可能である。また、実施例1、2と同様にCM9は停電等の事故により、制御情報が消失しないようにバッテリーでバックアップされ、しかも、二重化されている方が望ましい。また、NDM30も二重化されている方が信頼性が高くなるため望ましい。

【0040】なお、実施例1、2において本実施例で示したアドレスマップを使用して、分散化する制御情報かを判断することは可能である。また、DM11またはNDM30内に登録した制御情報を、同時にCM9にも登録することも可能である。このように、CM9にも同時に登録しておけば、DM11、NDM30内の登録された制御情報が消失しても、CM9内の制御情報を利用することが可能である。

【0041】(実施例4)本実施例では、制御情報高信頼性の目的からCM9、NDM30は二重化・不揮発化されているが、DM11は二重化されていない揮発性メモリであることを前提とする。本実施例では、上記前提の下で、CBS8のアクセス時間と制御情報の信頼性を実施例3と同等に保ったまま、不揮発分散化制御情報の一部をDM11に格納することを可能とした例を示す。

【0042】一般にNDM30の様な二重・不揮発メ

11

モリは、DM11の様な一重・揮発性メモリと比較して、実現コストが高い。そのためNDM30の容量はなるべく少ないことが望ましい。不揮発分散化制御情報の一部をDM11内に格納できれば、NDM30の必要容量削減が可能となり、制御情報分散化の実現コストを低減できる。本実施例におけるDKC1内の構成は実施例3と同様に図4のようになっており、データ転送系の動作は実施例1と同様である。また制御系の動作の中で揮発分散化制御情報と非分散化制御情報の操作は実施例3と同様である。そこで、制御系の動作の中で不揮発分散化制御情報の操作を中心に図4、図8、図9を用いて以下に説明する。

【0043】DM11には通常、揮発分散化制御情報のみを格納することが可能である。しかし不揮発分散化制御情報の場合も、CM9に同一の不揮発分散化制御情報が存在していれば、DM11に格納することが可能となる。何故ならば停電等の事故によりDM11の内容が消失しても、不揮発化されたCM9上に当該制御情報は残存するからである。そこでCM9に同一の不揮発分散化制御情報が存在する不揮発分散化制御情報はDM11に格納する。またCM9に同一の不揮発分散化制御情報が存在しない不揮発分散化制御情報はNDM30に格納する。不揮発分散化制御情報はDM11とNDM30の両方に同時存在することはない。以下図8は不揮発分散化制御情報参照要求(81)に対する処理フロー、図9は不揮発分散化制御情報登録要求(91)に対する処理フローである。それぞれについて以下に説明する。

【0044】参照がヒットした場合、自DM11あるいは自NDMに対して参照が行われるためCBS8アクセスは発生しない(83)。参照ミスの場合、CM9参照時間相等のCBSアクセスが発生する(84)、(85)。他DM11にも他NDM30にも同一制御情報が無い場合、CM9より自DM11へ参照制御情報が格納される(87)。他DM11に同一制御情報が存在する場合、他DM11より自DM11へ参照制御情報が格納される(89)。他NDM30に同一制御情報が存在する場合、他NDM30から他DM11へ参照情報が移動され(810)、CM9に登録が行われ(811)、他DM11より自DM11へ参照制御情報が格納される(89)。他NDM30から他DM11への参照情報移動はCBS8を介さない。またCM9への登録は、他DM11より自DM11への格納と同時に行為れるので、バスアクセス時間は長くない。

【0045】自NDM30に登録ヒットした場合と登録ミス時CM9にしか同一制御情報が無い場合、自NDM

(7)

特開平7-302171

12

30に登録が行われ、CBS8アクセスは発生しない(94)。自DM11に登録ヒットし、しかも他DM11に同一制御情報が存在しない場合、自DM11の無効化が行われた後(96)自NDM30に登録が行われる(94)。これは制御情報が自DM11から自NDM30に移動したことを意味する。移動に際してCBS8アクセスは発生しない。自NDM30に登録ミスし、他DM11あるいは他NDM30に同一制御データが存在する場合、CM9登録時間相等のCBSアクセスが発生する(97)、(98)。他DM11あるいは他NDM30の無効化(99)、CM9への登録(910)、自DM11への登録(911)が行われる。上記の三つの処理はCM9登録時間の中で同時に行われるので、バスアクセス時間は長くない。

【0046】

【発明の効果】従来のDKCでは制御情報を集中管理していたため、制御情報へのアクセスがネックになる危険性があった。しかし、本発明は制御情報の分散化を可能としたため、制御情報へのアクセスがネックになる危険性が無くなったため、性能向上を図ることが可能となった。

【図面の簡単な説明】

【図1】本発明の第1の実施例のディスク制御装置の構成を示すブロック図。

【図2】本発明の第1の実施例の処理フローチャート。

【図3】本発明の第2の実施例の処理フローチャート。

【図4】本発明の第3の実施例のディスク制御装置の構成を示すブロック図。

【図5】本発明の第3の実施例の処理フローチャート。

【図6】本発明の第3の実施例の制御情報のアドレスマップ図。

【図7】従来のディスクシステムの構成を示すブロック図。

【図8】本発明の第4の実施例の処理フローチャート。

【図9】本発明の第4の実施例の処理フローチャート。

【符号の説明】

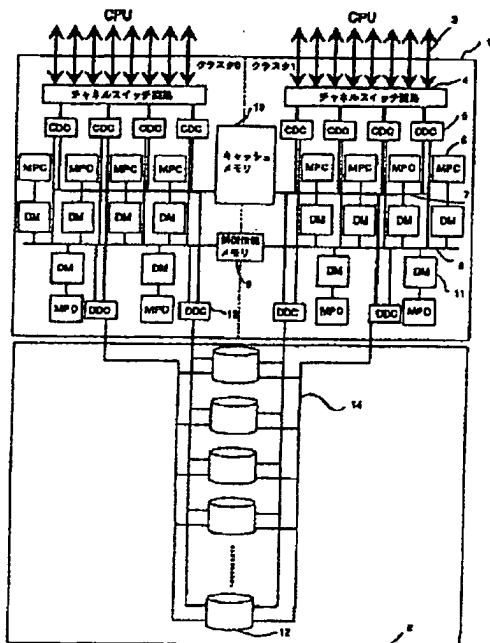
1…ディスク制御装置(DKC)、3…チャンネルバス、4…チャンネルスイッチ回路、5…チャンネルデータ制御回路(CDC)、6…マイクロプロセッサ(MP)、7…データ用共通バス(DBS)、8…制御情報用共通バス(CBS)、9…制御情報メモリ(CM)、10…キャッシュメモリ、11…分散メモリ(DM)、12…ドライブ、13…ドライブデータ制御回路(DDC)、14…ドライブバス、30…不揮発分散メモリ(NDM)。

(8)

特開平7-302171

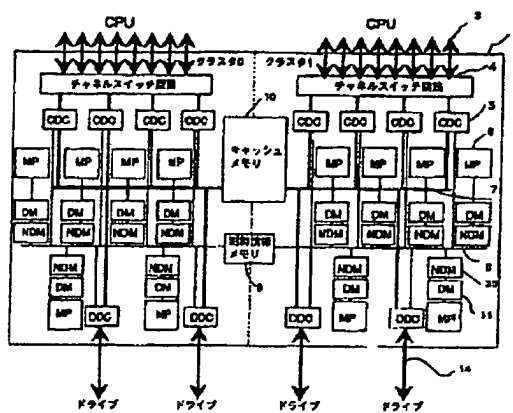
【图 1】

1



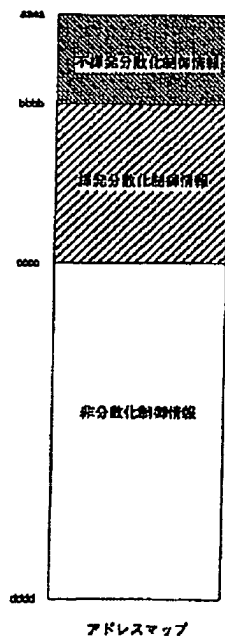
【圖4】

24



【圖6】

FIG 6

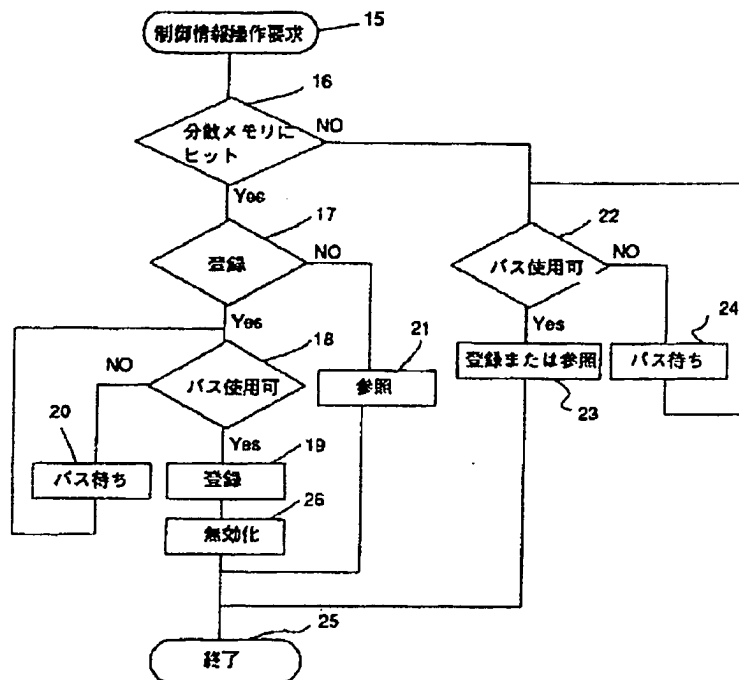


(9)

特開平7-302171

【図2】

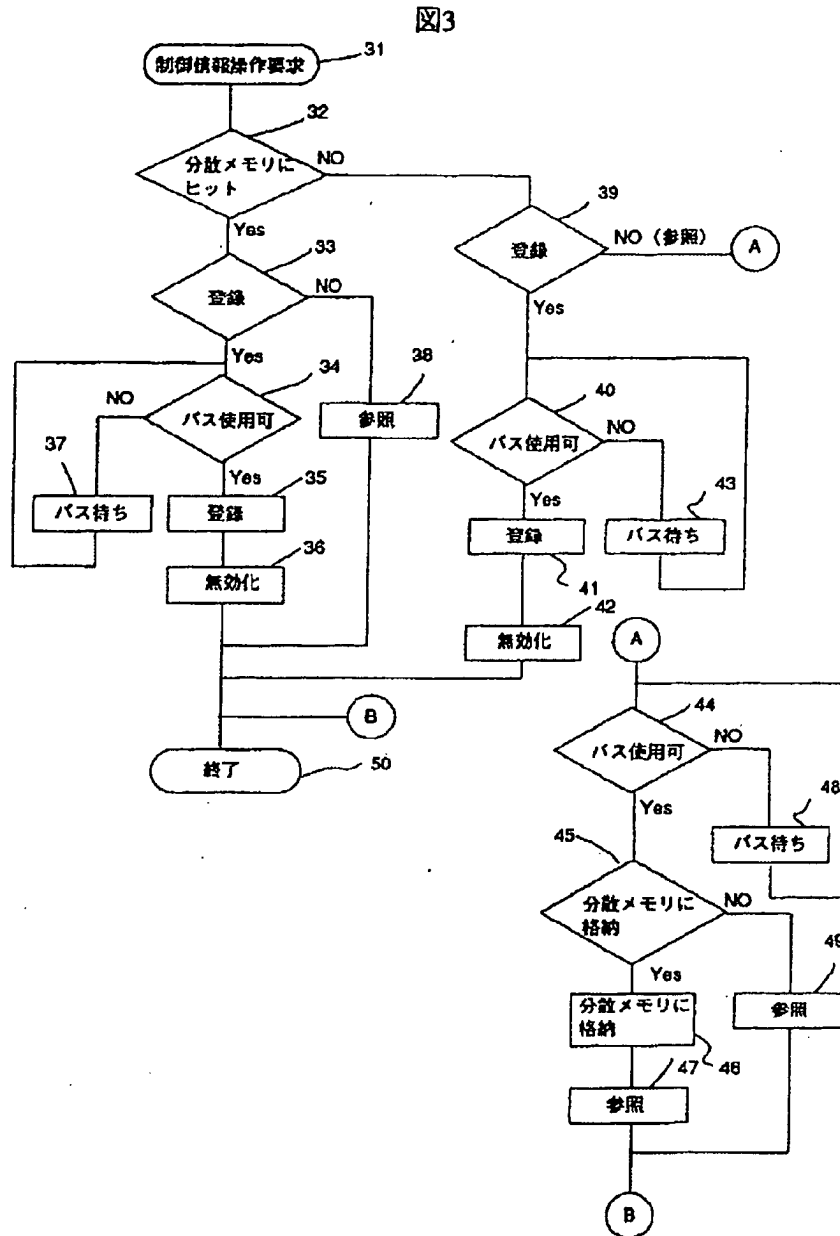
図 2



(10)

特開平7-302171

〔図3〕

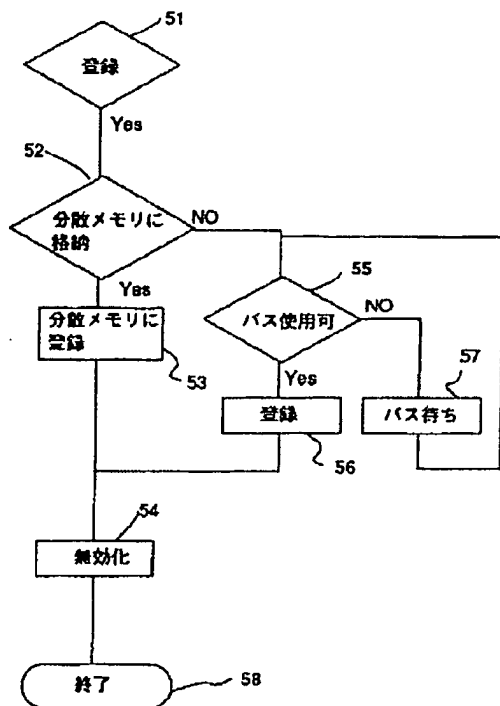


(11)

特開平7-302171

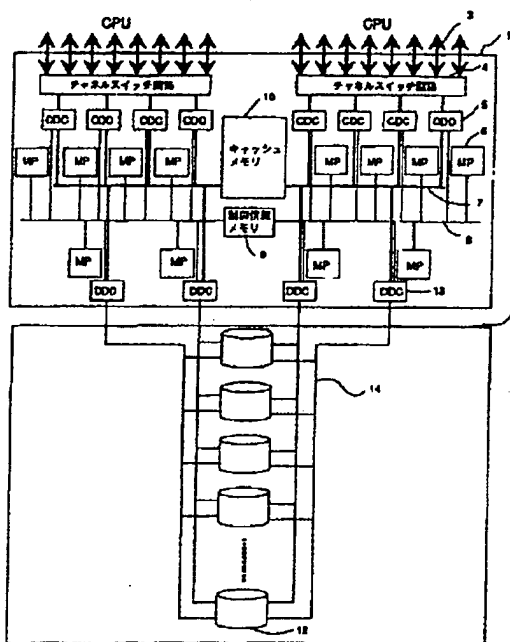
【図5】

図5



【図7】

図7

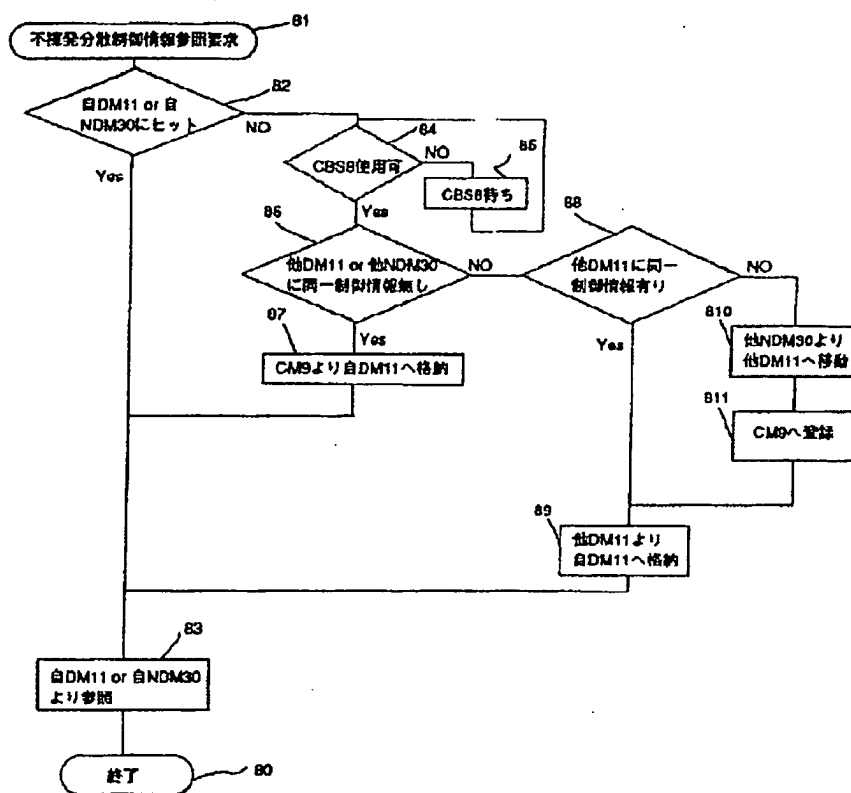


(12)

特開平7-302171

【図8】

図8

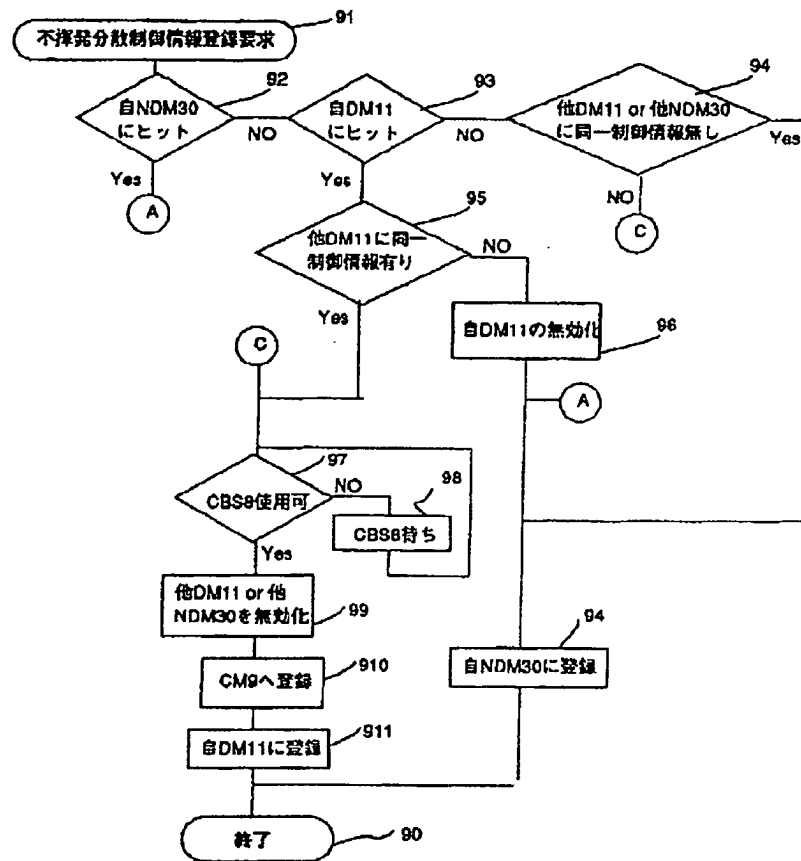


(13)

特開平7-302171

【図9】

図9



Reference 3

Japanese Patent Application Public-disclosure No. 7-302171
Japanese Patent Application Public-disclosure date: November
14, 1995

Title of the invention: Disk controller
Japanese Patent Application No. 6-96055
Japanese Patent Application date: May 10, 1994

Applicant: Hitachi Ltd.

Inventors: Tsunetaka Komachiya, Hitoshi Tsunoda and Yoshihisa
Kamo

An example of a disk controller (hereafter referred to as DKC) will be described below. As is described in Fig. 7, channel data control circuits (CDC) 5 and drive data control circuits (DDC) 13 conduct data transfer in DKC 1. The channel data control circuits (CDC) 5 conduct data transfer between CPUs and cache memory 10 in DKC 1, whereas the drive data control circuits (DDC) 13 conduct data transfer between the cache memory 10 and drives 12. The CDCs 5, cache memory 10 and DDCs 13 are connected to common bus (DBS) 7 for data. Data transfer control in the DKC 1 is controlled by microprocessors (MP) 6, which control the CDCs 5 and DDCs 13. The MPs 6 are each coupled to the common bus (CBS) 8 for control signals.

As is described above, the MPs 6 in the DKC 1 are each configured to have a multi-processor constitution, wherein the MPs 6 each efficiently utilize the common resources such as the buses, cache memory 10, drives 12 or the like to thereby facilitate smooth data transfer. Thus, in order for the MPs 6 to operate in concert with one another, control information in the control information memory (CM) 9 in the DKC 1 has to be repeatedly referred to and updated via the CBS 8. In a conventional DKC 1, control information is all gathered in the CM 9 and managed as a single unit.

[Problems to be solved by the invention]

Since the MPs 6 operate in concert in the conventional

DKC 1, the CM 9 needs to be accessed many times via the CBS 8 for single input/output processing. Therefore, if the number of the MPs 6 in the DKC 1 increases, or processing performance of the MPs 6 improves, access bottleneck arises between the CBS 8 and the CM 9.

[Means for solving the problems]

With a view to solving the aforementioned problem, the present invention provides each MP 6 in the disk controller with distribution memory (DM) 11 to distribute control information in the CM 9.

[Operation]

Thus, when each MP 6 is provided with DM 11 to distribute data in the CM 9, if control information that MP 6 requires is in the DM 11, the CM 9 no longer needs to be accessed, whereby access bottleneck between the CBS 8 and CM 9 is overcome.

[Embodiment]

(Embodiment 1)

An embodiment of the present invention will be described with reference to Fig. 1. Fig. 1 describes a configuration of the inside of the DKC 1. As is illustrated in Fig. 1, in the DKC 1, the cluster 0 and the cluster 1 are symmetric with respect to the cache memory 10 and control information memory 9. The DKC 1 consists of a data transfer system for conducting data transfer between the CPUs and drives 12 and a control system for managing the data transfer. Since the clusters 0 and 1 have exactly the same configuration, only the cluster 0 will be described below.

First, the data transfer system in the DKC 1 will be described. First, it will be explained how data is transferred between the CPUs and the DKC 1. The CPUs are coupled to the DKC 1 via the channel buses 3, which are coupled to the channel switch circuit 4 in the DKC 1. Since the maximum number of channel buses 3 that can operate simultaneously in the DKC 1 is four,

the channel switch circuit 4 selects up to four from more than four channel buses 3. The channel switch circuits 4 are respectively coupled to four channel data control circuits (CDC) 5. The CDCs 5 are coupled to the data common bus (DBS) 7, which is coupled to the cache memory 10.

Data transferred from the CPUs goes through the DBS 7 to be stored in the cache memory 10 under control of the CDCs 5. Thus, the CDCs 5 are circuits for controlling data transfer between the CPUs and the DKC 1.

Next, it will be explained how to transfer data between the DKC 1 and the drives 12. Not only the CDCs 5 and cache memory 10 but also two drive data control circuits (DDC) 13 are coupled to the DBS 7. Data transferred from the drives 12 passes through the DBS 7 to be stored in the cache memory 10 under control of the DDCs 13. Further, when data is stored in the drive 12 from the cache memory 10, the data is first read out of the cache memory 10 to pass through the DBS 7 and get transferred to the drive 12 under control of the DDCs 13. Thus, the DDCs 13 are circuits for controlling data transfer between the CPUs and the drives 12. As is described above, in the DKC 1, the CDCs 5 and DDCs 13 conduct data transfer between the CPUs, cache memory 10 and the drives 12.

Next, a control system for controlling the aforementioned data transfer will be described. In each cluster in the DKC 1, six microprocessors (MP) 6 are coupled to the common bus (CBS) 8 for control signals via distribution memory (DM) 11. Further, the control information memory (CM) 9 is coupled to the CBS 8. The MPs 6 are classified into two types, i.e., MPCs for controlling the CDCs 5 and MPDs for controlling the DDCs 13.

The MPCs are each assigned to control the specific CDCs 5. More specifically, the MPCs refer to control information and instruct the CDCs 5 on the basis of the result of the reference to conduct data transfer between the CPUs and the cache memory 10.

More specifically, as is described in Fig. 2, when the

CPU issues a data write request to the DKC 1, the MPC that received the request refers to the control information (15) to see whether or not data transfer to the cache memory 10 is possible. At this time, the MP1 checks (16) whether the control information is stored in the DM 11. If it transpires that the control information is stored in the DM11 of the MPC (hit), the control information in the DM 11 is referred to (21).